#100 DAYS OF RTL

https://github.com/Bhuvan-2602 www.LinkedIn.com/bhuvan-p-4825a9358

Description:

A Ripple Carry Adder adds two 4-bit binary numbers using a cascade of Full Adders. Each adder passes its carry to the next higher bit.

Objective:

Understand multi-bit binary addition and the concept of carry propagation in sequential adder design.

Inputs:

- a: 4-bit binary input
- b: 4-bit binary input
- cin: 1-bit initial carry input

Outputs:

- sum: 4-bit output representing the sum of inputs and carry
- cout: 1-bit final carry output

Design Approach:

- Cascade four Full Adders.
- Each full adder computes sum and carry.
- Carries ripple from least significant bit (LSB) to most significant bit (MSB).

Verilog Code:

Ripple Carry Adder

```
module ripple_carry_add(
  input [3:0] a,
  input [3:0] b,
  input cin,
  output [3:0] sum,
  output cout,
```

```
    wire c1, c2, c3
);
    full_add f1 (a[0], b[0], cin, sum[0], c1);
    full_add f2 (a[1], b[1], c1, sum[1], c2);
    full_add f3 (a[2], b[2], c2, sum[2], c3);
    full_add f4 (a[3], b[3], c3, sum[3], cout);
endmodule
```

Full Adder

```
module full_add(
    input a,
    input b,
    input c,
    output sum,
    output cout
);
    assign sum = a ^ b ^ c;
    assign cout = (a & b) | (b & c) | (c & a);
endmodule
```

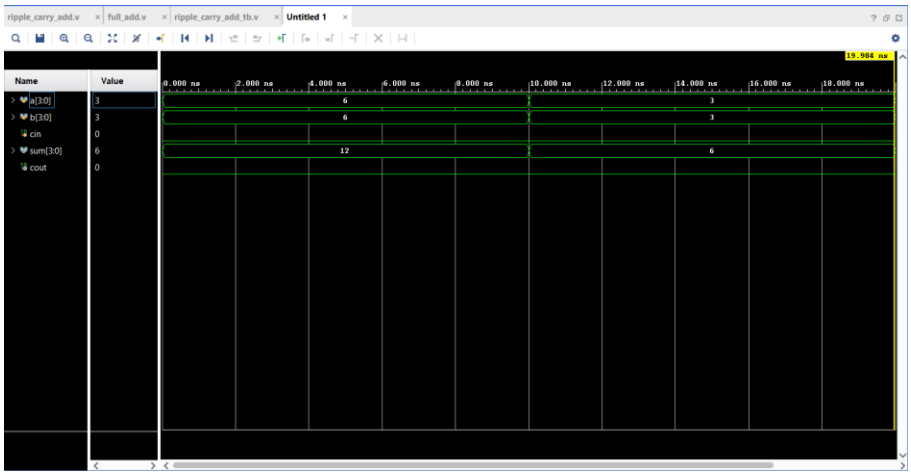
Testbench:

```
module ripple_carry_add_tb;
    reg [3:0] a, b;
    reg cin;
    wire [3:0] sum;
    wire cout;

    ripple_carry_add ut (a, b, cin, sum, cout);

    initial begin
        cin = 1'b0;
        a = 4'b0110; b = 4'b0110;  // Test case 1
        #10;
        a = 4'b0011; b = 4'b0011;  // Test case 2
        #10;
        $finish();
    end
endmodule
```

Waveform:



Schematic: